Interview Summary	Application No.	Applicant(s)	
	09/919,185	BODEN, EDWARD B.	
	Examiner	Art Unit	
	Victor Lesniewski	2152	

All participants (applicant, applicant's representative, PTO personnel):

(1) *Victor Lesniewski*. (3)_____.

(2) *Attorney Shelley Beckstrand, Reg. No. 24886*. (4)_____.

Date of Interview: *13 December 2006*.

Type: a)☒ Telephonic b)☐ Video Conference
c)☐ Personal [copy given to: 1)☐ applicant 2)☐ applicant's representative]

Exhibit shown or demonstration conducted: d)☐ Yes e)☒ No.
If Yes, brief description: _____.

Claim(s) discussed: *Allowable subject matter present in claims 2 and 10*.

Identification of prior art discussed: *Lucovsky (U.S. Patent Number 6,868,450) and Jackowski et al. (U.S. Patent Number 6,141,686)*.

Agreement with respect to the claims f)☒ was reached. g)☐ was not reached. h)☐ N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: *See Continuation Sheet*.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

BUNJOB JAROENCHONWANIT
SUPERVISORY PATENT EXAMINER
Examiner's signature, if required

Examiner Note: You must sign this form unless it is an Attachment to a signed Office action.

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record
A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews
Paragraph (b)
In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.
All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:
- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:
1) A brief description of the nature of any exhibit shown or any demonstration conducted,
2) an identification of the claims discussed,
3) an identification of the specific prior art discussed,
4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
5) a brief identification of the general thrust of the principal arguments presented to the examiner,
   (The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
6) a general indication of any other pertinent matters discussed, and
7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

Continuation of Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: The examiner has pointed to allowable subject matter present in claims 2 and 10. The applicant has agreed to amend all the independent claims to include either the subject matter of claim 2 or the subject matter of claim 10, either of which would make the independent claims allowable. The applicant has submitted a "Proposed Amendment" in which the independent claims have been amended and the examiner has agreed that all independent claims now contain allowable subject matter and that this case is now in condition for allowance. The applicant's response entitled "Proposed Amendment" is included herein where the new listing of claims is present on pages 2-48 and the applicant's remarks appear on pages 49-50.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of

Applicant	:	Edward B. Boden
Serial No.	:	09/919,185
Filed	:	30 July 2001
Examiner	:	Victor D. Lesniewski
Art Unit	:	2155
Entitled	:	System and Method for IP Packet Filtering Based on Non-IP Packet Traffic Attributes
Docket No.	:	END920010019US1

PROPOSED AMENDMENT

Honorable Commissioner
for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The following proposed amendment is submitted for approval and entry by Examiner's Amendment.

In the Claims

The status of claims in the case is as follows:

1. [Currently amended] A method for control and management of communication traffic, comprising the steps of:

expressing access rules as filters referencing system kernel data;

for outbound processing, determining source application indicia;

for inbound packet processing, executing a look-ahead function to determine target application indicia; said look-ahead function being executed within an IP layer of a protocol stack including ~~an IP layer~~ said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said

inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; [[and]]

responsive to said source or target application indicia, executing filter processing; said filter processing including constructing and evaluating logical expressions including non-IP packet attributes of arbitrary length, and selectively using a set of logical operators, alternative filter selector fields, and value set; and

executing said determining and executing steps within a kernel filtering function upon encountering a filter selector field referencing kernel data not included in said packet.

2. [Canceled]

3. [Previously presented] The method of claim 1, wherein said protocol stack is a TCP/IP protocol stack, and said filter processing including the steps of:

determining a task or thread identifier;

based on said task or thread identifier, determining a process or job identifier; and

based on said process or job identifier, determining job or process attributes for filter processing.

4. [Previously presented] The method of claim 1, wherein said protocol stack is a TCP/IP protocol stack, and said filter processing including the steps of:

determining a user identifier; and

based on said user identifier, determining user attributes for filter processing.

5. [Original] The method of claim 3, further comprising the step of determining from said task identifier a work

control block containing said process or job identifier.

6. [Canceled]

7. [Canceled]

8. [Previously presented] The method of claim 1, wherein said protocol stack is a TCP/IP protocol stack, and further comprising the steps of:

delivering to said filters infrastructure access rules for defining security context.

9. [Original] The method of claim 8, said infrastructure including logging, auditing, and filter rule load controls.

10. [Currently amended] A method for control and management of aspects of communication traffic within filtering, comprising the steps of:

receiving IP packet data into a TCP/IP protocol stack executing within a system kernel;

for an inbound IP packet, executing a look-ahead function within an IP layer of a protocol stack including ~~an IP layer~~ said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

executing filtering code within said IP layer of said system kernel with respect to non-IP packet data accessed within said system kernel outside of said TCP/IP protocol stack; said filtering code constructing and evaluating logical expressions of arbitrary length, and selectively using a set of logical operators, alternative filter selector fields, and value set.

11. [Original] The method of claim 10, said non-IP packet data including context data regarding said IP packet.

12. [Original] The method of claim 10, said non-IP packet data including data specific to a task generating said non-IP packet data.

13. [Original] The method of claim 10, said non-IP packet data including data specific to a task that will receive said IP packet.

14. [Original] The method of claim 11, said context data including packet arrival interface indicia.

15. [Canceled]

16. [Canceled]

17. [Canceled]

18. [Currently amended] A method for centralizing system-wide communication management and control within filter rules, comprising the steps of:

providing filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values;

for an inbound packet, executing a look-ahead function within an IP layer of a protocol stack including ~~an IP~~ said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered by said sockets layer;

said selector referencing data that does not exist in IP packets;

processing said filter statements, including constructing and evaluating logical expressions of

arbitrary length including non-IP packet attributes, and selectively using a set of logical operators, alternative filter selector fields, and value set;

executing said look-ahead function and processing said filter statements within a kernel filtering function upon encountering a filter selector field referencing kernel data not included in said packet.

19. [Previously presented] The method of claim 18, wherein said protocol stack is a TCP/IP protocol stack, and said parameters selectively including userid, user profile, user class, user group, user group authority, user special authority, job name, process name, job group, job class, job priority, other job or process attributes, and date & time.

20. [Previously presented] The method of claim 18, wherein said protocol stack is a TCP/IP protocol stack, and said filters statements being provided within a user interface to said system.

21. [Previously presented] The method of claim 18, wherein said protocol stack is a TCP/IP protocol stack, and further comprising the steps of:

establishing a tunnel between two IP address limiting traffic to applications bound to ports at each end of said tunnel;

said filtering code accessing filtering attributes further limiting traffic selectively to job indicia; and

operating said filtering code within a kernel filtering function upon encountering a filter selector field referencing kernel data not included in said traffic.

22. [Currently amended] A method for traversing a portion only of a protocol stack to disallow selective IP packet traffic, comprising the steps of:

receiving a packet in the system kernel of the

operating system of a first node from an application, said kernel including a filter processor; said filter processor for constructing and evaluating logical expressions of arbitrary length including non-IP packet attributes, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set;

for inbound packet processing to a first node from a second node, executing a look-ahead function in an IP layer of ~~the system~~ said system kernel of said first node to determine a target application; said system kernel including a TCP/IP protocol stack including ~~an IP layer~~ said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer upon encountering a filter selector field referencing kernel data not included in said inbound packet provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer source application indicia identifying the application layer application to which said packet

would have been delivered;

for both said inbound packet processing, and for outbound packet processing from said first node to said second node, executing within said kernel the steps of

processing said packet by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

determining a user ID, process or job identifier from said work control block;

from the user ID, process or job identifier selectively determining attributes for said user process or job; and

passing said attributes to said filter processor for managing and controlling

communication traffic.

23. [Currently amended] A method for expressing access rules as filters, comprising the steps of:providing a filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values; and

said selector referencing data within a system kernel outside of a protocol stack that does not exist in IP packets for controlling access to an application;

for an inbound IP packet, executing a look-ahead function within the IP layer of ~~a protocol~~ said protocol stack, said protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said look-ahead function in said IP layer upon encountering a filter selector field referencing kernel data not included in said inbound IP packet provides to said transport layer said inbound IP

packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

processing said filter statements within said IP layer of said protocol stack with respect to non-IP packet data accessed within said system kernel outside of said protocol stack by constructing and evaluating logical expressions including non-IP packet attributes of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set referencing said application layer application.

24. [Currently amended] A method for managing and controlling communication traffic by centralizing access rules in filters including non-IP packet attributes executing within and referencing data available in system kernels outside of a protocol stack having an IP layer, a transport layer, and a sockets layer, comprising the steps

for outbound packet processing from a first node to a second node of:

receiving said packet in the kernel of the operating system of said first node from an application or process at said first node;

processing said packet by determining a task ID; responsive to said task ID, determining a corresponding work control block;

responsive to said work control block, determining a process or job identifier;

responsive to said process or job identifier, determining job or process attributes; and

executing said filters within said IP layer with respect to non-IP packet data accessed within said system kernel outside of said protocol stack by constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively

including a set of logical operators, alternative filter selector fields, and value set.

25. [Currently amended] The method of claim 24, further comprising the steps for inbound packet processing from said second node to said first node of:

initially operating said kernel at said first node upon encountering a filter selector field referencing kernel data not included in said inbound packet to determine a target application for said inbound packet at said first node by executing a look-ahead function within said IP layer of said [[a]] said protocol stack ~~including an IP layer, a transport layer, a sockets layer, and an application layer~~ and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered[[;]].

26. [Canceled]

27. [Canceled]

28. [Canceled]

29. [Currently amended] A method for managing and controlling communication traffic by centralizing the access rules, comprising the steps for outbound packet processing from a first node to a second node of:

receiving said packet in the system kernel of the operating system of said first node from an application or process at said first node, said kernel including a filter processor for constructing and evaluating logical expressions including non-IP packet attributes of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields referencing kernel data outside of a protocol stack, and value set, said protocol stack including an IP layer, a transport layer, a sockets layer, and an application layer;

processing said packet within said IP layer including referencing non-protocol stack portions of said system kernel ~~a TCP/IP stack~~;

by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

determining a user ID control block from said work control block;

from the user ID control block determining attributes for said user; and

passing said attributes to said filter processor for managing and controlling communication traffic.

30. [Currently amended] The method of claim 29, further comprising the steps for inbound packet processing from

said second node to said first node of:

initially operating said kernel at said first node to determine a target application for said packet at said first node by executing a look-ahead function within said IP layer of said TCP/IP protocol stack, said TCP/IP protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered.

31. [Canceled]

32. [Canceled]

33. [Canceled]

34. [Currently amended] A method for control and management of communication traffic with respect to a

system node, comprising the steps of:

receiving at said system node an inbound packet; [[and]]

executing within a protocol stack of the system kernel of said system node a filtering function identifying for said inbound packet a filter including non-IP packet attributes referencing non-packet data, and constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set; [[and]]

responsive to said filter, executing a look-ahead function for identifying a target application for said inbound packet; said look-ahead function executed within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-~~

~~deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

executing said filtering function and said look-ahead function within said kernel upon encountering a filter selector field referencing kernel data not included in said packet.

35. [Currently amended] The look-ahead function of the method of claim 34 wherein said protocol stack is a TCP/IP protocol stack, and further comprising the steps of:

passing to a transport layer function identified by an IP header a packet marked ~~non-deliverable~~ deny for determining which user-level process or job is to receive said packet;

receiving from said transport layer an application layer task identifier for said user-level process or

job; and thereafter

passing said packet marked by said task identifier to said transport layer for delivery to said application layer task.

36. [Currently amended] System for control and management of communication traffic, comprising:

a system kernel including a filter function and stack data;

said filter function including a filter <u>including non-IP packet attributes</u> selectively referencing said stack data for expressing access rules;

said filter function being responsive to receipt of an outbound packet for determining a source application;

said filter function being responsive to receipt of an inbound packet ~~processing~~ <u>including a filter selector field referencing kernel data not included in said</u>

inbound packet for executing a look-ahead function within the IP layer of a TCP/IP protocol stack to determine a target application; said protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

said filter function being responsive to said source or target application for executing filter processing including constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields referencing kernel data not included in a packet, and value set.

37. [Currently amended] A system for control and management of aspects of communication traffic within filtering, comprising:

a system kernel;

a protocol stack including an IP layer, a transport layer, a sockets layer, and an application layer for executing within said IP layer of said system kernel, responsive to an inbound IP packet, a look-ahead function by which said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

filtering code within said system kernel operable with respect to non-IP packet data accessed within said system kernel outside of said protocol stack for controlling and managing said aspects of communication traffic; said filter code for constructing and

evaluating logical expressions of arbitrary length including non-IP packet attributes, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set.

38. [Currently amended] A system for centralizing system-wide communication management and control within filter rules including non-IP packet attributes, comprising:

filter statements having a syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values;

said selector referencing data that does not exist in IP packets;

a look-ahead function within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer, which look-ahead function, responsive to

encountering a filter selector field in an inbound packet referencing kernel data not included in said ~~for an~~ inbound packet, executes within said IP layer ~~provides~~ to provide to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and ~~receives~~ receive back from said transport layer indicia, provided to said transport layer by said sockets layer, for identifying the application layer application to which said packet would have been delivered; and

a filter processor for constructing and evaluating filter statements including logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields selectively referencing non-IP packet data accessed within said system kernel outside of said protocol stack, and value set.

39. [Currently amended] A system for traversing a portion only of a TCP/IP protocol stack to disallow selective IP packet traffic, comprising:

a system kernel;

a filter processor executing within said system kernel for constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields selectively referencing non-packet data accessed within said system kernel outside of said TCP/IP protocol stack, and value set;

said filter processor responsive to an inbound packet for executing within an IP layer of said TCP/IP protocol stack a look-ahead function for determining a target application; said ~~look-ahead function operating within said~~ TCP/IP protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer; and which, for said [[IP]] inbound packet, upon encountering a filter selector field referencing kernel data not included in said inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-~~

~~deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;

said filter processor responsive to both inbound and outbound packets for

processing said packet by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

determining a user ID, process or job identifier from said work control block;

from the user ID, process or job identifier selectively determining attributes for said user process or job; and

passing said attributes to said filter

processor for managing and controlling communication traffic.

40. [Currently amended] A system for expressing access rules as filters, comprising:

filter statements for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values;

said selector referencing data that does not exist in IP packets for controlling access to an application;

a look-ahead function executing within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for an inbound packet upon encountering a filter selector field referencing kernel data not included in said inbound packet, said ~~IP layer~~ look-ahead function provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said

transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

a filter processor for constructing and evaluating said filter statements as logical expressions of arbitrary length, each said logical expression selectively including said operator selected from a set of logical operators, alternative filter selector fields including non-IP packet attributes, and value set.

41. [Currently amended] A system for managing and controlling communication traffic by centralizing access rules in filters including non-IP packet attributes executing within and referencing data available in system kernels, comprising:

a computer readable medium;

first code for receiving a packet in the kernel of the operating system of a first node from an application

or process at said first node; said kernel responsive to an inbound packet, for executing a look-ahead function within the IP layer of a TCP/IP protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, upon encountering a filter rule referencing kernel data not included in said inbound packet, said ~~IP layer~~ look-ahead function provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;

second code for processing said packet by determining a task ID;

third code responsive to said task ID for determining a corresponding work control block;

fourth code responsive to said work control block for

determining a process or job identifier;

fifth code responsive to said process or job identifier for determining job or process attributes;

sixth code for executing said filters by constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set; and wherein

said first, second, third, fourth, fifth, and sixth code is recorded on said computer readable medium.

42. [Canceled]

43. [Currently amended] A system for control and management of communication traffic with respect to a system node, comprising:

a filtering function executing within the IP layer of a protocol stack of the system kernel of said system

node identifying for an inbound packet a filter referencing non-packet data within said system kernel and outside of said protocol stack; [[and]]

a look-ahead function responsive to said filter referencing non-packet data within said system kernel and outside of said protocol stack for identifying a target application for said inbound packet; said look-ahead function functioning within said IP layer of [[a]] said protocol stack including [[an]] said IP layer, a transport layer, and a sockets layer, ~~and an application layer~~ and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the ~~application layer~~ application to which said packet would have been delivered;[[;]] and

a filter processor for constructing and evaluating logical expressions of arbitrary length, said logical

expressions selectively including a set of logical operators, alternative filter selector fields, and value set.

44. [Canceled]

45. [Currently amended] A computer program product for control and management of aspects of communication traffic within filtering, said computer program product comprising:

a computer readable medium;

first program instructions to receive IP packet data into a TCP/IP protocol stack executing within a system kernel including, for processing an inbound IP packet, a look-ahead function within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, upon encountering a filter selector field referencing kernel data not included in said inbound IP packet, said IP layer provides to said transport layer said

inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;

second program instructions to execute filtering code within said system kernel with respect to non-IP packet data accessed within said system kernel outside of said TCP/IP protocol stack by constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set; and wherein

said first and second program instructions are recorded on said medium.

46. [Currently amended] A computer program product for centralizing system-wide communication management and control within filter rules, said computer program product comprising:

a computer readable medium;

first program instructions to execute filter statements including non-IP packet attributes having a syntax for accepting parameters in the form of a selector, each selector specifying selector field, a logical operator selected from a set of a plurality of logical operators, and a set of values; and

second program instructions to cause said selector to reference data within an operating system kernel outside of a protocol stack and that does not exist in IP packets, said data including application layer indicia obtained for an incoming packet by a look-ahead function; said look-ahead function executing within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, upon encountering a selector field referencing kernel data not included in [[for]] said IP inbound packet, said ~~IP layer~~ look-ahead function provides to said

transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and wherein

said first and second program instructions are recorded on said medium.

47. [Currently amended] A computer program product for managing and controlling communication traffic by centralizing access rules in filters including non-IP packet attributes executing within and referencing data available in system kernels, said computer program product comprising:

a computer readable medium;

first program instructions to receive said packet in the kernel of the operating system of said first node from a process at said first node;

second program instructions to process said packet by determining a task ID;

third program instructions, responsive to said task ID, to determine a corresponding work control block;

fourth program instructions, responsive to said work control block, to determine a process or job identifier;

fifth program instructions, responsive to said process or job identifier, to determine job or process attributes; and

sixth program instructions to execute a filter processor within the IP layer of a protocol stack for constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields referencing non-IP packet attributes for accessing data within said system

kernels and outside of said protocol stack, and value set; and wherein

said first, second, third, fourth, fifth, and sixth program instructions are recorded on said medium.

48. [Currently amended] The computer program product of claim 47, wherein said protocol stack is a TCP/IP protocol stack, and said computer program product further comprising for inbound packet processing from said second node to said first node:

sixth program instructions to initially operate said kernel at said first node to determine a target application for said packet at said first node by executing a look-ahead function within the IP layer of a protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to

said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;; and wherein

said sixth program instructions are recorded on said medium.

49. [Currently amended] A computer program product for control and management of communication traffic, comprising:

a computer readable medium;

first program instructions for expressing access rules as filters <u>including non-IP packet attributes</u> referencing system kernel data <u>outside of a protocol stack</u>;

second program instructions, for outbound processing, for determining a source application;

third program instructions, for inbound packet

processing, for executing a look-ahead function to determine a target application; said look-ahead function operating within the IP layer of ~~a protocol~~ said protocol stack, said protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, upon encountering a filter selector field referencing kernel data not included in an inbound packed, said ~~IP layer~~ look-ahead function provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;

fourth program instructions, selectively responsive to said source and target ~~application~~ applications, and upon encountering a filter selector field referencing kernel data not included in said inbound packet for executing filter processing including constructing and evaluating logical expressions of arbitrary length,

said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set; and wherein

said first, second, third, and fourth program instructions are recorded on said computer readable medium.

50. [Currently amended] A computer program product for control and management of aspects of communication traffic within filtering, comprising:

a computer readable medium;

first program instructions for receiving IP packet data into a TCP/IP protocol stack including an IP layer executing within a system kernel;

second program instructions for executing filtering code within said IP layer of said system kernel with respect to non-IP packet data accessed within said system kernel outside of said TCP/IP protocol stack;

said filtering code constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields including non-IP packet attributes, and value set; and wherein

said first and second program instructions are recorded on said computer readable medium.

51. [Currently amended] A computer program element for centralizing system-wide communication management and control within filter rules, comprising:

a computer readable medium;

first program instructions for providing filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, a logical operator, and a set of values,

second program instructions for executing filtering by

constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including said logical operator selected from a set of logical operators, at least one said selector field including non-IP packet attributes accessed within a system kernel and outside of a protocol stack, and at least one said value;

said selector referencing data that does not exist in IP packets including data obtained, for an inbound IP packet, by executing a look-ahead function within the IP layer of said protocol stack, said [[a]] protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer, and which, for said IP inbound packet, upon encountering a selector field referencing kernel data not included in said inbound IP packet, said ~~IP layer~~ look-ahead function provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application

to which said packet would have been delivered; and wherein

said first and second program instructions are recorded on said computer readable medium.

52. [Currently amended] A computer program product for managing and controlling communication traffic by centralizing access rules in filters on non-IP packet attributes executing within, and referencing data available in, system kernels, comprising:

a computer readable medium;

first program instructions for receiving said packet in the system kernel of the operating system of said first node from an application or process at said first node;

second program instructions for processing said packet by determining a task ID;

third program instructions, responsive to said task ID, for determining a corresponding work control block;

fourth program instructions, responsive to said work control block, for determining a process or job identifier;

fifth program instructions, responsive to said process or job identifier, for determining job or process attributes;

sixth program instructions for executing a filter processor within the IP level of a protocol stack with respect to non-IP packet data accessed within said system kernel outside of said protocol stack for constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set; and wherein

said first, second, third, fourth, fifth, and sixth

program instructions are recorded on said computer readable medium.

53. [Currently amended] The computer program product of claim 52, further comprising for inbound packet processing from said second node to said first node:

seventh program instructions initially operating said kernel at said first node to determine a target application for said packet at said first node by executing a look-ahead function within said IP layer of said [[a]] protocol stack, said protocol stack including [[an]] said IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as ~~non-deliverable~~ deny, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and wherein

said seventh program instructions are recorded on said computer readable medium.

REMARKS

Claims 1-5, 8-14, 18-25, 29-30, 34-41, 43, and 45 -53 are in the case, none as yet allowed.

Interview

Applicant's attorney expresses appreciation for courtesy extended by Examiner Victor Lesniewski in a telephone interview on 13 Dec 2006. The above amendment incorporates suggestions made by Examiner Victor Lesniewski to incorporate material from claims 2 and 10, which are deemed to contain allowable subject matter when combined with the changes submitted in the Amendment After Final submitted on or about 17 Nov 2006.

SUMMARY

Applicants urge that claims 1-5, 8-14, 18-25, 29-30,

34-41, 43, and 45 -53 be allowed as set forth in the above proposed amendment.

Sincerely,

Edward B. Boden

By

_____

Shelley M Beckstrand
Reg. No. 24,886

Date: 13 Dec 2006

Shelley M Beckstrand, P.C.
Patent Attorney
61 Glenmont Road
Woodlawn, VA 24381-1341

Phone: (276) 238-1972
Fax: (276) 238-1545
Correspondence Address:

IBM Corporation
Intellectual Property Law (Dept. 917, Bldg. 006-1)
3605 Highway 52 North
Rochester, MN 55901-7829